논문 20-45-03-17 The Journal of Korean Institute of Communications and Information Sciences '20-03 Vol.45 No.03
https://doi.org/10.7840/kics.2020.45.3.584

Kolmogorov-Smirnov 검정을 이용한 인터리버 파라미터 블라인드 추정

김 윤 지•, 김 근 배*, 정 운 섭**, 윤 동 원°

Blind Interleaver Parameter Estimation Using Kolmogorov-Smirnov Test

Yoonji Kim•, Geunbae Kim*, Unseob Jung**, Dongweon Yoon°

요 약

본 논문에서는 송신단에서 컨볼루션 부호화기와 블록 인터리버를 사용하는 시스템에서, 수신된 데이터만을 이용하여 인터리버의 주기를 블라인드로 추정하는 알고리즘을 제안한다. 먼저 블록 인터리빙 된 컨볼루션 부호어로 구성된 행렬의 랭크에 대해 살펴보고 이를 이용하여 수신된 데이터로 이루어진 정방 행렬의 랭크 부족(Rank deficiency) 확률 분포와 랜덤 이진 정방 행렬의 랭크 부족 확률 분포를 콜모고로프 스미르노프 검정(Kolmogorov-Smirnov test; KST)으로 비교함으로써 인터리버 주기를 추정한다. 제안한 알고리즘의 추정 성능을 이진 대칭 채널(Binary symmetric channel; BSC)에서 모의 실험을 통해 구하고 주기 검출 확률을 이용하여 기존 알고리즘의 성능과 비교 분석한다.

Key Words : Blind estimation, Kolmogorov-Smirnov Test, Block Interleaver, Convolutional Code

ABSTRACT

In this paper, we propose a blind estimation algorithm for interleaver period by utilizing received data especially when the transmitter in a communication system includes a convolutional encoder and a block interleaver. We first investigate the rank of the matrices composed of block interleaved convolutional codewords. And then, we compare rank deficiency distribution of matrix composed of received data with that of matrix composed of randomly generated binary data by using Kolmogorov-Smirnov Test (KST) to estimate the interleaver period. The estimation performance in the binary symmetric channel(BSC) is analyzed through computer simulations and compared with that of the previous estimation algorithm in terms of detection probability.

Ⅰ. 서 론

통신의 신뢰도를 높이기 위해 대부분의 통신 시스템에서는 채널에서 발생하는 랜덤 오류를 정정할 수 있는 오류 정정 부호(Error correction code)를 사용한다. 그러나 채널에서 연집 오류(Burst error)가 발생한

※ 이 연구는 방위사업청 및 국방과학연구소의 재원에 의해 설립된 신호정보 특화연구센터 사업의 지원을 받아 수행되었음.
• First Author : Hanyang University Department of Electronics and Computer Engineering, natasha0826@hanyang.ac.kr, 학생회원
° Corresponding Author : Hanyang University Department of Electronics and Computer Engineering, dwyoon@hanyang.ac.kr, 종신회원
* Hanyang University, gbkim@hanyang.ac.kr
** The 2nd R&D Institute - 2nd Directorate, ADD, jeus@add.re.kr, 정회원
논문번호 : 201911-279-0-SE, Received October 31, 2019; Revised December 24, 2019; Accepted December 27, 2019

다면 오류 정정 부호를 사용한 오류 정정이 어려워지며 복호화에 치명적인 영향을 끼치게 된다. 따라서 복호화 성능을 저하시키는 연집 오류를 랜덤 오류(Random error)로 분산시키기 위해 일정 주기로 특정 패턴을 이용하여 데이터의 송신 순서를 바꿔주는 인터리버가 오류 정정 부호와 함께 사용된다[1].

일반적인 통신 시스템에서는 인터리버의 파라미터를 송신단과 수신단이 공유하여 디인터리빙이 가능하지만 수신단이 송신단과 전송 파라미터를 공유하지 않는 블라인드 통신 환경에서는 수신단에서 수신 데이터 비트들만으로 인터리버의 파라미터를 추정해서 디인터리빙을 수행해야 한다. 특히 인터리빙이 일정 주기로 수행되므로 인터리버의 파라미터 가운데 인터리버 주기 추정이 우선적으로 수행되어야 한다.

최근까지 인터리버 주기를 블라인드로 추정하기 위한 다양한 방법들이 연구되어 왔다[2-8]. 이러한 추정 방법들은 송수신단에서 사용한 오류 정정 부호의 종류에 따라 크게 메시지가 블록 부호화 되었을 경우[4,5]와 컨볼루션 부호화 되었을 경우[6,7]로 분류가 가능하다.

메시지가 블록 채널 부호화 된 후 블록 인터리빙 된 경우의 추정 알고리즘으로 잡음이 있는 채널에서 랜덤 이진 행렬의 랭크 부족 분포와 수신 데이터 행렬의 랭크 부족 분포를 비교하여 오류가 적게 포함된 벡터들을 선별하는 알고리즘이 제시되었다[4]. 이후에는 두 확률 분포의 차이를 통계적 기법으로 정량화하고 오경보를 제어하는 알고리즘이 제안되었다[5].

메시지가 컨볼루션 채널 부호화 된 후 블록 인터리빙 된 경우의 인터리버 파라미터 추정 알고리즘에는 인터리빙된 컨볼루션 부호어의 패리티 검사식을 이용하는 알고리즘[6]과 선형성이 남아있는 것으로 추정되는 열의 개수로 데이터 행렬의 랭크와 랭크 부족을 추정하는 알고리즘[7]이 제안되었다. 또한 미지의 부호어가 블록 인터리빙 되었을 때 송신단에서 사용한 오류 정정 부호를 구분하고 인터리버의 파라미터를 추정하는 알고리즘[8]이 제안되었다. 그러나 기존의 알고리즘들[7],[8]은 오류가 포함되어있는 비트의 행렬 내의 위치에 따라 열 벡터들의 특성을 잘못 판단할 수 있다는 문제점이 존재한다.

본 논문에서는 송신단에서 메시지 비트들이 컨볼루션 부호화 되고 블록 인터리빙 되었을 때, 부호화 과정 중 발생하는 부호어 내의 선형성을 이용하여 인터리버 주기를 추정하는 알고리즘을 제안하고 추정 성능을 분석한다. 이를 위해 수신 데이터로 구성한 정방 행렬에 컨볼루션 부호의 선형 특성이 나타나는 조건에 대해 분석하고, 이를 바탕으로 랭크 부족에 대한 확률 질량 함수 (Probability mass function)를 KST의 검정 통계량으로 비교하여 인터리버 주기를 추정한다. 또한 이진 대칭 채널에서 모의 실험을 통하여 주기 추정 확률을 구하고 인터리버 추정 성능을 분석한다.

Ⅱ. 인터리버 추정 알고리즘

이 장에서는 블록 인터리빙 된 컨볼루션 부호어로 이루어진 정방 행렬의 랭크에 대해 분석하고 이를 이용한 블록 인터리버의 주기 추정 알고리즘을 제안한다.

2.1 시스템 모델

인터리버 파라미터 추정을 위하여 가정한 시스템 모델은 그림 1과 같다. 메시지 비트들은 (n,k,K) 컨볼루션 부호화 되고 행의 수가 N_r, 열의 수가 N_c인 (N_r,N_c) 블록 인터리버를 거쳐 인터리빙 된다. 여기서 n과 k는 각각 부호어 길이와 부호화기의 입력 비트 수를, K는 구속장 길이를 나타내며 블록 인터리버 주기 β는 $N_r \times N_c$ 다.

인터리버의 출력 비트 열 y는 채널을 통과하면서 오류 w가 더해지게 되고 수신단에서는 y와 다른 비트 열 s를 수신하게 된다. 여기서 y, w, s는 GF(2)의 정수로 0 또는 1이다. 본 논문에서는 송신단에서 사용한 전송 파라미터에 관한 어떠한 정보 없이 수신 데이터 비트 열 s만을 이용하여 인터리버의 주기 β를 추정하는 알고리즘을 제시한다. 본 논문에서 β는 부호어 길이 n의 정수배로 가정한다.

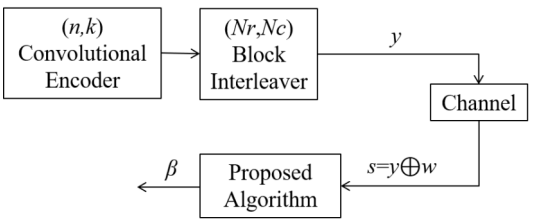

그림 1. 시스템 모델
Fig. 1. System model

2.2 컨볼루션 부호의 선형성을 이용한 인터리버 주기 추정

이 절에서는 오류가 없는 수신 데이터 비트들로 구성한 $a \times a$ 정방 행렬의 랭크와 인터리버 주기 사이의 관계를 알아본다. 수신 데이터로 구성된 행렬은 행렬

의 크기와 인터리버 주기에 따라 랭크가 달라진다.

따라서 수신 비트들로 구성한 행렬의 랭크는 인터리버 주기 추정의 중요한 단서가 된다. 이를 조사하기 위해 먼저 인터리빙 되지 않은 부호어로 구성한 행렬의 랭크를 살펴본 후 이러한 행렬과 인터리빙 된 부호어로 구성한 행렬 간 관계에 대해 분석한다.

2.2.1 인터리빙 되지 않은 컨볼루션 부호어로 구성한 행렬

인터리빙 되지 않은 컨볼루션 부호화기의 출력 비트 열로 이루어진 $a \times a$ 정방 행렬을 만들었다고 가정했을 때, 이 행렬의 행 벡터 간 선형성이 존재한다면, 이 행렬의 랭크는 a보다 작은 값을 가지며 이 행렬은 랭크 부족 행렬이 된다.

그림 2는 초기 값이 M_0, M_{-1}, M_{-2}이고 생성 다항식이 $g^v(x) = g_1^v + g_2^v x + g_3^v x^2 (v = 1,2)$인 (2,1,3) 컨벌루션 부호화기를 나타낸다. 여기서 $g^v(x)$들은 서로소이다. 이 부호화기에 서로 독립적인 메시지 비트 M_1, M_2, M_3… 이 입력되어 컨볼루션 부호어 $C_t = \{C_t^1, C_t^2\}$가 출력된다면 C_t는 다음과 같이 현재의 입력 비트와 레지스터에 저장된 이전 2개의 입력 비트의 배타적 논리 합으로 나타낼 수 있다.

$$C_t^1 = g_1^1 M_t \oplus g_2^1 M_{t-1} \oplus g_3^1 M_{t-2} \tag{1}$$

$$C_t^2 = g_1^2 M_t \oplus g_2^2 M_{t-1} \oplus g_3^2 M_{t-2} \tag{2}$$

여기서 C_t는 t번째 부호어를 나타내며 $C_t^v (v = 1,2)$는 부호화기의 v번째 다항식에 의해 생성된 출력 비트이다.

정방 행렬의 열 또는 행의 크기 a가 n의 배수인 경우와 그렇지 않은 경우에 행렬의 랭크가 각각 달라지는데, 이에 대하여 살펴보도록 한다.

(1) a가 n의 배수인 경우

a가 n의 배수일 때 행렬의 랭크를 분석하기 위해 그림 3과 같이 C_t로 만들어진 8×8 행렬을 예시로 든다.

이 행렬의 랭크를 조사하기 위해 이 행렬의 각 행 벡터 R_1, R_2,...,R_8을 식 (1)과 (2)를 이용하여 전개하면 다음 식들과 같이 구할 수 있다.

식 (3), (4), (5)를 통해 8개의 행 벡터 R_1, R_2,...,R_8는 모두 표 1과 같이 6개의 기저(Basis) 벡터들의 선형 조합으로 표현이 가능함을 알 수 있다.

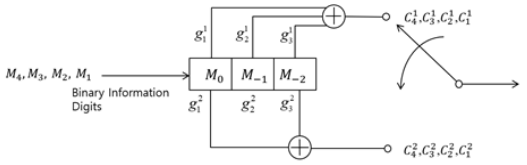

그림 2. (2,1,3) 컨볼루션 부호화기
Fig. 2. (2,1,3) Convolutional encoder

$$\begin{aligned} R_1 &= (C_1^1, C_5^1, C_9^1, C_{13}^1, C_{17}^1, C_{21}^1, C_{25}^1, C_{29}^1) \\ &= g_1^1(M_1, M_5, M_9, M_{13}, M_{17}, M_{21}, M_{25}, M_{29}) \\ &\oplus g_2^1(M_0, M_4, M_8, M_{12}, M_{16}, M_{20}, M_{24}, M_{28}) \\ &\oplus g_3^1(M_{-1}, M_3, M_7, M_{11}, M_{15}, M_{19}, M_{23}, M_{27}) \end{aligned} \tag{3}$$

$$\begin{aligned} R_2 &= (C_1^2, C_5^2, C_9^2, C_{13}^2, C_{17}^2, C_{21}^2, C_{25}^2, C_{29}^2) \\ &= g_1^2(M_1, M_5, M_9, M_{13}, M_{17}, M_{21}, M_{25}, M_{29}) \\ &\oplus g_2^2(M_0, M_4, M_8, M_{12}, M_{16}, M_{20}, M_{24}, M_{28}) \\ &\oplus g_3^2(M_{-1}, M_3, M_7, M_{11}, M_{15}, M_{19}, M_{23}, M_{27}) \end{aligned} \tag{4}$$

...

$$\begin{aligned} R_8 &= (C_4^2, C_8^2, C_{12}^2, C_{16}^2, C_{20}^2, C_{24}^2, C_{28}^2, C_{32}^2) \\ &= g_1^2(M_4, M_8, M_{12}, M_{16}, M_{20}, M_{24}, M_{28}, M_{32}) \\ &\oplus g_2^2(M_3, M_7, M_{11}, M_{15}, M_{19}, M_{23}, M_{27}, M_{31}) \\ &\oplus g_3^2(M_2, M_6, M_{10}, M_{14}, M_{18}, M_{22}, M_{26}, M_{30}) \end{aligned} \tag{5}$$

행렬의 랭크는 행 공간(Row space)의 기저 벡터의 수와 같으므로 그림 3의 행렬의 랭크는 6이 된다. 마찬가지로 C_t로 구성된 a=10인 10×10 행렬의 랭크는 7이 된다. 이처럼 a가 n의 배수일 때 기저의 수가 a가 아닌 까닭은 현재의 n개의 출력 비트는 $\frac{a}{n}k$ 개의 현재 입력 비트와 K-1개의 이전 입력 비트의 조합으로

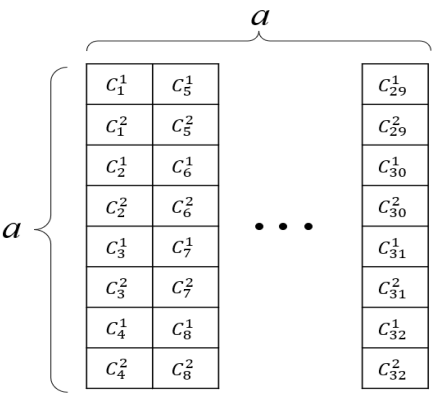

그림 3. 컨볼루션 부호어로 구성된 정방 행렬
Fig. 3. A square matrix composed of convolutional codewords

얻어지므로 행 벡터를 구성하는 기저 벡터 중 겹치는 벡터들이 존재하기 때문이다. 즉, a가 n의 배수라면 $a \times a$ 행렬은 랭크 부족 행렬이 되며 행렬의 랭크는

$$\begin{aligned} Rank &= \frac{a}{n}kK - (\frac{a}{n}k-1)(K-1) \\ &= \frac{a}{n}k + K - 1 \end{aligned} \tag{6}$$

이 된다.

(2) a가 n의 배수가 아닌 경우

a가 n의 배수가 아닌 행렬의 행 벡터들은 그림 3의 행렬의 행 벡터들과 달리 하나의 행 벡터를 이루는 모든 출력 비트들이 하나의 생성 다항식으로부터 생성된 것이라고 할 수 없다. 따라서 각각의 행 벡터를 식 (3), (4), (5)와 같이 전개하여 기저 벡터를 구하면 a개의 기저 벡터가 얻어 지므로 행렬의 랭크는 a가 된다.

이처럼 인터리빙 되지 않은 컨볼루션 부호어로 행렬을 구성하면 a가 n의 배수인 경우에만 컨볼루션 부호어 내의 선형 특성이 행렬에 나타나게 되며 이로 인해 행렬은 랭크 부족 행렬이 된다.

2.2.2 인터리빙 된 컨볼루션 부호어로 구성한 행렬

여기서는 인터리빙 된 컨볼루션 부호어들로 구성한 행렬과 인터리빙 되지 않은 행렬간의 관계에 대해 알아보고 이를 이용하여 수신 비트로 구성한 행렬의 랭크를 구한다. 앞에서와 마찬가지로 s로 정방 행렬을 구성한다고 하였을 때 행렬의 랭크는 a가 β의 배수인 경우와 그렇지 않은 경우에 따라 달라진다.

(1) a가 β의 배수인 경우

그림 4는 C_t가 주기 8인 (2,4) 블록 인터리빙 되었을 때 구성한 8×8 행렬을 나타낸다. 그림 3과 그림 4

표 1. 그림 2의 행렬의 기저 벡터
Table 1. Basis for row vectors of the matrix in Fig 2

Basis
$(M_{-1}, M_3, M_7, M_{11}, M_{15}, M_{19}, M_{23}, M_{27})$
$(M_0, M_4, M8, M_{12}, M_{16}, M_{20}, M_{24}, M_{28})$
$(M_1, M_5, M_9, M_{13}, M_{17}, M_{21}, M_{25}, M_{29})$
$(M_2, M_6, M_{10}, M_{14}, M_{18}, M_{22}, M_{26}, M_{30})$
$(M_3, M_7, M_{11}, M_{15}, M_{19}, M_{23}, M_{27}, M_{31})$
$(M_4, M_8, M_{12}, M_{16}, M_{20}, M_{24}, M_{28}, M_{32})$

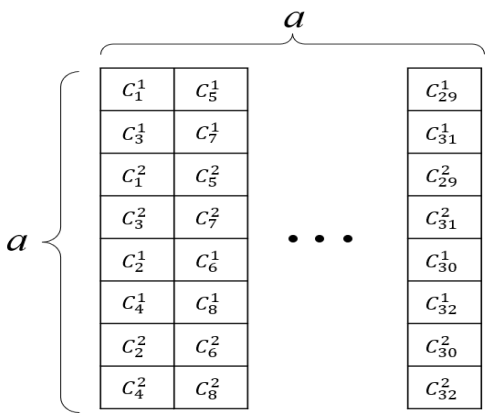

그림 4. (2,4) 블록 인터리빙 된 컨볼루션 부호어로 구성한 정방 행렬
Fig. 4. A square matrix composed of (2,4) block interleaved convolutional codewords

의 행렬을 비교해 보면, 그림 4의 행렬은 그림 3의 행렬에 단순히 행 교환(Row permutation)이 일어난 것임을 알 수 있으며, 행 교환은 행렬의 랭크에 영향을 주지 않으므로 그림 4의 행렬의 랭크는 식 (6)과 같게 된다.

(2) a가 β의 배수가 아닌 경우

C_t가 주기가 6인 (3,2) 블록 인터리빙 되었을 때 C_t로 구성한 8×8행렬을 그림 5에 도시하였다. 그림 4와 달리 그림 5의 행렬은 행 크기 8이 인터리버 주기 6의 배수가 아니므로 그림 3의 행렬에서 같은 행에 있던 C_t^v가 다른 행으로 흩어져 배치된다. 이렇게 흩어진 C_t^v가 새로운 하나의 행 벡터를 이루게 되기 때문에 그림 5의 행렬의 행 벡터들을 식 (3), (4), (5)와 같이 전개하여 기저 벡터를 조사하면 2.2.1.2의 행렬과 마찬가지로 8개의 기저를 구할 수 있어, 행렬의 랭크는 8이 된다.

즉, C_t가 블록 인터리빙 되었다면 a가 인터리버 주기 β의 배수일 때만 행렬의 랭크가 식 (6)과 같으며 그렇지 않은 경우, 행렬의 랭크는 a와 같다. 이를 통해 수신된 비트들로 구성한 행렬이 랭크 부족 행렬이 되는 경우는 행의 수 a가 β의 배수임을 알 수 있다.

2.3 확률 분포의 비교를 통한 인터리버 주기 추정 알고리즘

인터리버의 주기를 추정하기 위해 수신 데이터 비트 열 s의 길이를 L이라고 할 때, s를 길이가 a인 벡터

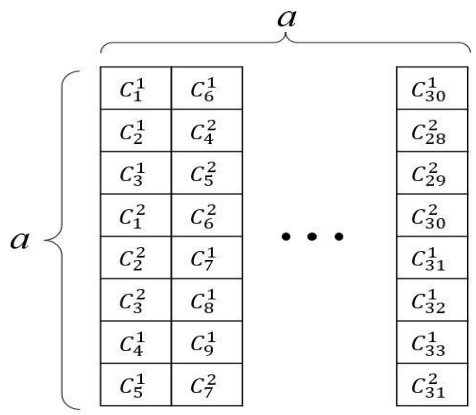

그림 5. (3,2) 블록 인터리빙 된 컨볼루션 부호어로 구성한 정방 행렬
Fig. 5. A square matrix composed of (3,2) block interleaved convolutional codewords

s_i($i=1,2,...,\left\lfloor \frac{L}{a} \right\rfloor$, $\lfloor x \rfloor$ 는 x이하의 정수) 로 나눈 후 이 벡터들 중 임의로 a개를 선택하여 $a\times a$ 정방 행렬 H를 만든다. 2.2 절의 결과에 의하여 $a=\lambda\beta$ (λ는 자연수)일 때 H의 랭크는 식 (6)과 같으므로 H는 랭크 부족 행렬이 된다. $a=\lambda\beta$ 일 때 H가 랭크 부족 행렬이라면 a가 $(\lambda+1)\beta$일 때의 H 또한 랭크 부족 행렬이므로 연속된 두 랭크 부족 행렬의 행 또는 열의 길이 a의 차는 β와 같다[7]. 반대로 $a\neq\lambda\beta$이면 H는 랜덤하게 발생된 0과 1로 이루어진 행렬로 볼 수 있다[3]. 본 논문에서는 $a_{min}\leq a\leq a_{max}$ 일 때 H의 랭크 부족 확률 분포를 통해 H가 랭크 부족 행렬인지 또는 랜덤하게 발생된 비트로 이뤄진 행렬인지 선별하고 이를 이용해 인터리버 주기를 추정한다.

2.3.1 랜덤 이진 정방 행렬의 랭크 부족 확률 분포

이론적인 랜덤 이진 정방 행렬의 랭크 부족 확률 분포에 의하면, 랭크 부족을 t라 할 때, t=1일 확률이 가장 높으며 t=0, 2, 3 순으로 발생 확률이 높다[9]. 따라서 H의 랭크가 우연히 a보다 3이상 작을 확률은 매우 낮다. 따라서 만약 H의 랭크가 a-3 이하라면 H는 2.2절에 설명한 부호화 과정에서 발생하는 행 벡터 간 선형성으로 인한 랭크 부족 행렬이라 볼 수 있으며, 이 행렬을 구성하는 비트에 오류가 덜 포함되어 있다고 가정할 수 있다[4]. 이 가정이 성립하기 위해선 오류가 없을 때 H의 랭크가 a-3보다 작거나 같아야 한다. 즉, $a \geq \frac{n(K+2)}{n-k}$이어야 한다. 따라서 a_{min} 의 값은

$$\therefore a_{min} = \left\lceil \frac{n(K+2)}{n-k} \right\rceil \quad (7)$$

과 같게 된다.

2.3.2 KS검정을 이용한 주기의 배수 추정

KS 검정을 이용하여 인터리버의 주기를 추정하기 위하여 본 논문에서는 우선 오류가 덜 포함된 벡터를 선별하는 알고리즘을 이용하여 인터리버 주기 추정에 사용하게 될 벡터 집합 V를 구성한다[4]. 이렇게 구성된 집합 V에서 다시 a개의 벡터를 임의로 선택하여 $a\times a$ 행렬 $\tilde{H}$ 생성하고 랭크 부족을 구한다. 이 과정을 N번 반복 시행하면 $\tilde{H}$에 대한 랭크 부족 확률 분포를 구할 수 있다.

$\tilde{H}$ 의 랭크 부족 확률 분포와 랜덤 이진 정방 행렬의 랭크 부족 확률 분포를 비교하고 두 분포의 차이 ρ가 가장 큰 m개의 a에 대하여 집합 A를

$$A=\{a_1, a_2, \ldots, a_m\} \ (m\geq 2\text{인 자연수}) \quad (8)$$

로 정의한다. 여기서 a_i(i=1,2, ⋯, m)는 두 확률 분포의 차이가 i번째로 큰 경우의 a이다. ρ는 KST의 검정 통계량으로

$$\rho = max|F_{\tilde{H}}(x) - F_R(x)| \quad (9)$$

로 정의된다[10]. 여기서$F_{\tilde{H}}(x)$는 $\tilde{H}$의 랭크 부족 누적 질량 함수로 $F_{\tilde{H}}(x)=\frac{n(x)}{N}$이며, $n(x)$는 N번의 시행 중 랭크 부족이 x이하인 행렬 $\tilde{H}$의 수이다. $F_R(x)$는 랜덤 이진 행렬의 랭크 부족 누적 질량 함수로 표 2와 같다[8].

만약 $a\neq\lambda\beta$ 라면 $\tilde{H}$의 랭크 부족 누적 확률 분포는 랜덤 이진 정방 행렬의 누적 확률 분포와 유사하므로 ρ는 0에 가까운 작은 값이 된다. 그러나 $a=\lambda\beta$인 경우 $\tilde{H}$는 랭크 부족 행렬로 랜덤 이진 정방 행렬의 랭크

표 2. 랜덤 이진 정방 행렬의 누적 확률 질량 함수
Table 2. Cumulative probability mass function of random binary square matrix

x	$F_R(x)$
0	0.288788
1	0.866364
2	0.994714
a	1

부족 확률 분포와는 다른 랭크 부족 확률 분포를 가지게 되므로 ρ는 상대적으로 큰 값이 된다. 따라서 구성된 집합 A는 β의 배수로 이루어진 집합이다.

2.3.3 KLD를 이용한 주기의 배수 검증

추정의 정확도를 향상시키고 오경보 확률을 낮추기 위하여 두 랭크 부족 확률 분포를 KLD를 이용하여 한 번 더 비교한다. 비교를 위해 행렬의 행의 수가 a_i일 때의 랭크 부족 확률 분포와 랜덤 이진 정방 행렬의 랭크 부족 확률 분포 간 쿨백-라이블러 발산(Kullback-Leibler Divergence, KLD)인 D_{KL}를 구한다. D_{KL}은

$$D_{KL} = \sum_{x} P(x) \log \frac{P(x)}{Q(x)} \tag{10}$$

로 정의된다[11]. 여기서 $P(x)$는 랜덤 이진 정방 행렬의 랭크 부족 확률 분포, $Q(x)$는 $\widetilde{H}$의 랭크 부족 확률 분포이다. D_{KL}은 항상 0보다 크거나 같은 값이며 두 확률 분포가 유사할수록 0에 가까운 작은 값을 갖고 두 확률 분포가 다를수록 큰 값을 갖는다. 만약 랜덤 이진 정방 행렬의 랭크 부족 확률 분포와 $\widetilde{H}$의 랭크 부족 확률 분포의 D_{KL}이 특정 임계값 이하인 경우 $\widetilde{H}$는 랜덤 이진 행렬의 랭크 부족 확률분포와 유사하다고 볼 수 있다[5]. 따라서 만약 D_{KL}이 임계값 이하일 때의 a_i는 주기의 배수가 아니므로 A의 원소에서 제외한다.

KST와 KLD를 통하여 β의 배수로 검증된 A의 원소 a_i들을 크기순으로 정렬하면 인접한 a_i들 간 차의 최소는 β와 같게 되어 인터리버의 주기의 추정이 가능하다.

Ⅲ. 모의 실험 및 성능 분석

3.1 수신 데이터로 구성한 행렬과 랜덤 이진 정방 행렬의 랭크 부족 확률 분포

a가 인터리버 주기의 배수일 때의 랭크 부족 확률 분포와 랜덤 이진 정방 행렬의 랭크 부족 확률 분포를 실험을 통해 비교하고 그 결과를 그림 6에 도시하였다. 표 3에 표기된 파라미터들은 수신 데이터에 오류가 존재하는 환경에서 a가 주기의 배수일 때의 $\widetilde{H}$의 랭크 부족 분포와 랜덤 이진 정방 행렬의 랭크 부족 분포의 차이를 검증하기 위한 하나의 예로 실험에 사용한 시뮬레이션 파라미터이며 그림 6의 별표는 $\widetilde{H}$의 랭크 부족 확률 분포이며 실선은 랜덤 이진 행렬의 이론적인 랭크 부족 확률 분포를 나타낸다.

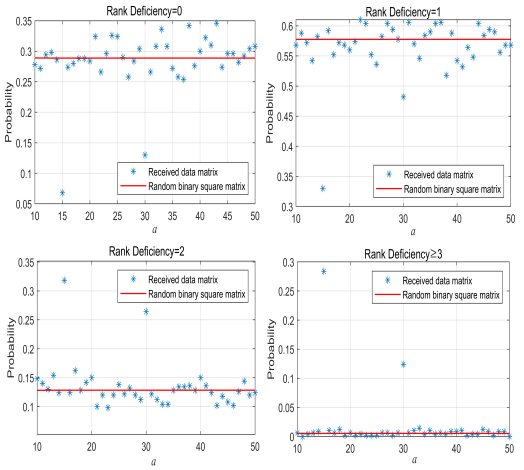

그림 6. $\widetilde{H}$와 랜덤 이진 정방 행렬의 랭크 부족 확률 분포
Fig. 6. Probability distributions of the rank deficiency of $\widetilde{H}$ and random binary square matrix

그림 6에서 a가 인터리버 주기의 배수인 15, 30인 경우, $\widetilde{H}$의 랭크 부족 확률 분포가 랜덤 이진 행렬의 랭크 부족 확률 분포와 큰 차이를 보이는 반면 a가 주기의 배수가 아닌 경우에는 그 분포가 랜덤 이진 행렬의 분포와 유사하다. 예외적으로 a가 45인 경우, 45가 주기의 배수임에도 행렬의 랭크 부족 확률 분포가 랜덤 이진 정방 행렬과 차이를 보이지 않는다. 이는 행렬의 크기가 클수록 행렬 안에 오류가 더 많이 포함되어있기 때문이다.

표 3. 시뮬레이션 파라미터
Table 3. Simulation parameters

Parameters	Value
Channel	BSC
Error probability	0.04
Convolutional code	(3,1,4)
Interleaving period	15

3.2 추정 성능 분석

이 절에서는 송수신단에 (3,1,4) 컨볼루션 부호와 (3,1,7) 컨볼루션 부호가 사용되었을 때 BSC에서 제안한 방법의 추정 성능을 모의 실험을 통해 구하고 이를 [7]의 알고리즘과 비교 분석한다.

본 논문에서는 [7]의 알고리즘과 성능을 비교하기 위하여 컨볼루션 부호로 [7]에서 성능을 제시한 (3,1,4)와 (3,1,7) 부호를 사용하였으나, 다른 컨볼루션

부호에 대해서도 마찬가지로 적용이 가능하다.

그림 7과 그림 8에는 각각 입력 비트들이 생성 다항식의 계수가 $[13,15,17]_8$, $[133,165,171]_8$인 (3,1,4), (3,1,7) 컨볼루션 부호화된 후 주기가 12인 블록 인터리버로 인터리빙 되었을 때, 제안한 알고리즘과 [7]의 인터리버 주기 추정 확률을 채널의 비트 오류 확률(Bit error rate, BER)에 따라 나타내었다. [7]은 수신 데이터로 열의 수가 행의 수의 정수 배인 직사각 행렬을 구성해 인터리버 주기를 추정한다. 이 때 행렬의 행 벡터의 길이가 길수록 선형성을 지닌 벡터를 더 정확하게 선별할 수 있기 때문에 추정 성능이 더 좋아지지만 행렬을 구성하는데 필요한 데이터의 수가 많아진다는 단점이 있다. 모의 실험을 통하여 제안한 알고리즘과 [7]의 알고리즘의 추정 성능 비교를 위해 [7]에서 행렬의 열의 수가 행의 수의 2배라고 가정하였다.

그림 7에서 볼 수 있듯이 (3,1,4) 컨볼루션 부호가 사용된 경우, 제안한 방법은 BER이 0.077일 때 90%의 추정 확률을 보인 반면, [7]은 BER이 0.04일 때 90%의 추정 확률을 보이고 있다. 한편 (3,1,7) 부호가 사용된 경우, 그림 8과 같이 제안한 방법은 BER이 0.037일 때, [7]은 BER이 0.031일 때 각각 90%의 추정 확률을 보인다. 위의 두 실험 결과들을 통해 제안하는 알고리즘은 채널상황이 더 좋지 않은 상황에서도 [7]과 같은 추정 성능을 얻을 수 있음을 알 수 있다. 즉, 제안한 방법이 [7]의 방법보다 동일한 BER에서 주기 추정 성능이 우수함을 알 수 있다. 이는 기존의 인터리버 추정 알고리즘은 수신 데이터 행렬을 이루는 열 벡터들의 특성을 이용한 반면, 제안한 방법은 확률 분포를 통해 통계적인 기법으로 주기를 추정하기 때문이다.

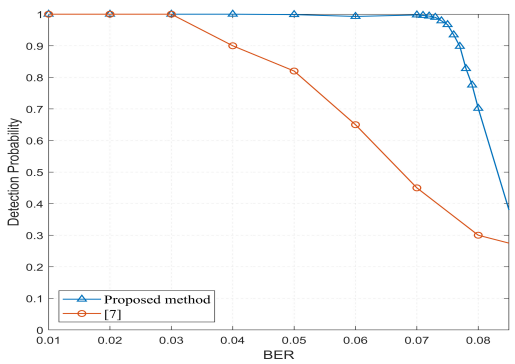

그림 7. (3,1,4) 컨볼루션 부호가 사용되었을 때 BSC에서의 주기 검출 확률
Fig. 7. Detection probability for interleaving period with (3,1,4) convolutional code over BSC

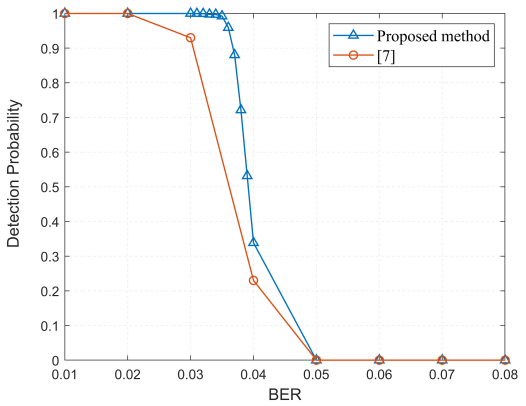

그림 8. (3,1,7) 컨볼루션 부호가 사용되었을 때 BSC에서의 주기 검출 확률
Fig. 8. Detection probability for interleaving period with (3,1,7) convolutional code over BSC

한편 송신단에서 같은 부호율의 오류 정정 부호를 사용하더라도 구속장 길이 K가 3일 때의 추정 성능이 K가 7일 때의 추정 성능보다 좋다. 이는 구속장의 길이가 클수록, 식 (6)에 의해 데이터로 구성된 행렬의 랭크 부족이 줄어들게 되면서 랭크 부족 행렬을 구별할 확률이 감소하기 때문이다.

Ⅳ. 결 론

본 논문에서는 송신단에서 컨볼루션 부호화되고 블록 인터리빙 되었을 때, 수신 데이터 비트들로 구성한 행렬의 랭크에 대해 분석하고 이를 이용하여 인터리버 주기를 블라인드로 추정하는 알고리즘을 제안하였다. 인터리빙된 컨볼루션 부호어로 구성한 정방 행렬의 랭크는 행렬의 행의 수가 인터리버 주기의 배수인 경우에만 사용된 컨볼루션 부호의 파라미터와 행렬의 크기에 따라 결정되었다. 따라서 정방 행렬의 행의 개수가 인터리버 주기의 배수일 때와 배수가 아닐 때의 행렬의 랭크 부족 분포는 서로 다른 분포를 나타내었다.

제안한 알고리즘은 오류가 덜 포함된 수신 데이터 벡터를 선별한 다음, 수신 비트 열로 이루어진 정방 행렬과 랜덤 이진 정방 행렬의 랭크 부족 분포를 KST의 검정 통계량으로 비교함으로써 인터리버 주기의 배수를 추정하였다. 또한 행의 수가 추정된 주기의 배수일 때의 행렬과 랜덤 이진 정방 행렬의 확률 분포를 KLD를 통해 다시 한 번 비교하여 임계값보다 작은 경우 최종적으로 인터리버 주기의 배수로 결정하였다.

모의 실험을 통해서 송신단에서 (3,1,4) 컨볼루션

부호와 (3,1,7) 컨볼루션 부호가 사용된 경우기존의 알고리즘보다 각각 오류 확률이 0.037, 0.006 높은 채널에서도 90% 이상의 검출 확률을 보였다.

향후에는 실제 사용 환경을 고려하여 인터리버 주기가 부호어의 배수가 아닌 경우 또는 수신 데이터의 수가 행렬을 구성하기에 충분하지 않은 경우 등 다양한 조건에서의 인터리버 파라미터 추정 연구가 진행되어야할 것이다.

References

[1] B. Sklar, "*Digital communications: Fundamentals and applications*," Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[2] G. Burel and R. Gautier, "Blind estimation of encoder and interleaver characteristics in a non-cooperative context," in *Proc. IASTED*, pp. 275-280, Scottsdale, AZ, Nov. 2003.

[3] G. Sicot, S. Houcke, and J. Barbier, "Blind detection of interleaver parameters," *Sign. Process.*, vol. 89, no. 4, pp. 450-462, Apr. 2009.

[4] C. Choi and D. Yoon, "Enhanced blind interleaver parameters estimation algorithm for noisy environment," *IEEE Access*, vol. 6, pp. 5910-5915, Sep. 2017.

[5] G. Kim, Y. Jang, and D. Yoon, "Blind estimation for interleaving parameter using probability mass function over fading channel," *J. KIIT*, vol. 17, no. 5, pp. 39-46, May 2019.

[6] A. Tixier, "Blind identification of an unknown interleaved convolutional code," in *Proc. ISIT 2015*, pp. 71-75, Hongkong, China, Jun. 2015.

[7] R. Swaminathan and A. S. Madhukumar, "Parameter estimation of block and helical scan interleavers in the presence of bit errors," *Digit. Sign. Process.*, vol. 60, pp. 20-32, Jan. 2017.

[8] R. Swaminathan and A. S. Madhukumar, "Classification of error correcting codes and estimation of interleaver parameters in a noisy transmission environment," *IEEE Trans. Broadcast.*, vol. 63, no. 3, Sep. 2017.

[9] V. F. Kolchin, "*Random graphs*," Cambridge Univ. Press, pp. 126-131, 1999.

[10] W. W. Daniel, "*Applied nonparametric statistics*," PWS-Kent, pp. 319-330, 1990.

[11] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79-86, Mar. 1951.

김 윤 지 (Yoonji Kim)

2019년 2월 : 한양대학교 융합전자공학부 학사 졸업

2019년 2월~현재 : 한양대학교 전자컴퓨터통신공학과 석박사과정

<관심분야> 블라인드 복조, 통신 공학, 신호 정보

[ORCID:0000-0003-4247-5795]

김 근 배 (Geunbae Kim)

1991년 2월 : 한양대학교 전자통신공학과(공학사)

1993년 2월 : 한양대학교 전자통신공학과(공학석사)

2012년 2월 : 한양대학교 전자통신전파공학과(공학박사)

2020년 1월 현재 : 한양대학교 산학협력단 연구 교수

<관심분야> 무선통신, 채널 코딩, 신호정보

[ORCID:0000-0003-3262-091X]

정 운 섭 (Unseob Jeong)

1988년 2월 : 충남대학교 전자공학과(공학사)

1990년 2월 : 충남대학교 전자공학과(공학석사)

2007년 2월 : 충남대학교 통신공학(공학박사)

2020년 1월 현재 : 국방과학연구소 전자전 기술부장

<관심분야> 전자전 시스템 설계 및 신호처리

[ORCID:0000-0002-0399-4317]

윤 동 원 (Dongweon Yoon)

1989년 2월 : 한양대학교 전자통신공학과(공학사)
1992년 2월 : 한양대학교 전자통신공학과(공학석사)
1995년 8월 : 한양대학교 전자통신전파공학과(공학박사)
2020년 1월 현재 : 한양대학교 융합전자공학부 교수

<관심분야> 무선통신, 위성 및 우주통신, 신호정보

[ORCID0000-0001-9631-3500]